1)Сделать так, чтобы при перебрасывании на модуль, к пути в строке браузера добавлялось название документа

1. Сделать при удалении модуля удаление всех связанных с ним задач. Добавить попап с подтверждением удаления

//Пример создания модуля

{

"name":"Модуль4",

"description":"Описание4",

"dateTimeStart":"2023-04-22T18:46:52.000Z",

"dateTimeEnd":"2023-04-22T18:46:52.000Z",

"responsible":"6453f7f5f96a77c38464920e"

}

//Пример создания задачи

    {

        "name": "Задача12",

        "description": "Описание12",

        "status": "Completed",

        "executor": "6453ef3ef94558aa04b7f2ca",

        "modulee": "6453f3cdc46a0bc44c5d2339"

}

//Пример создания субъекта

{

"fullname":{

"first\_name":"Ann",

"middle\_name":"",

"last\_name":"Davidchuk"

},

"email":"Davidchuk@mail.ru",

"group":"6454a1cf13681c3e308f8e70"

}

//Пример создания группы

{

"name": "Excluded",

"description": "Group Excluded"

}

const express = require('express');

const bodyParser = require('body-parser');

const mongoose = require('mongoose');

const http = require('http');

const cors = require('cors');

const { routes } = require('./src/routes');

const { join } = require('path');

//=========file

const multer = require('multer');

const { GridFsStorage } = require('multer-gridfs-storage');

const Grid = require('gridfs-stream');

const methodOverride = require('method-override');

//=========file/

// настроим подключение к бд

const mongoHost = '127.0.0.1';

const mongoPort = '27017';

const mongoDbname = 'some-mongo';

mongoose.connect(`mongodb://${mongoHost}:${mongoPort}`, {

  useCreateIndex: true,

  useNewUrlParser: true,

  useUnifiedTopology: true,

});

const app = express();

require('dotenv').config();

//=========file

let gfs;

const conn = mongoose.connection;

conn.once('open', () => {

  gfs = Grid(conn.db, mongoose.mongo);

  gfs.collection('uploads');

});

const storage = new GridFsStorage({

  url: `mongodb://${mongoHost}:${mongoPort}/${mongoDbname}`,

  file: (req, file) => {

    return new Promise((resolve, reject) => {

      const filename = file.originalname;

      const fileInfo = {

        filename: filename,

        bucketName: 'uploads',

      };

      resolve(fileInfo);

    });

  },

});

const upload = multer({ storage });

app.use(methodOverride('\_method'));

//=========file/

// инициализируем приложение

app.use(cors());

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.use(function (req, res, next) {

  res.header('Access-Control-Allow-Origin', '\*');

  res.header(

    'Access-Control-Allow-Headers',

    'Origin, X-Requested-With, Content-Type, Accept',

  );

  next();

});

routes.forEach((item) => {

  app.use(`/api/v1/${item}`, require(`./src/routes/${item}`));

});

//=========file

// Роут для загрузки файла

const uploadRoute = require('./src/routes/upload');

app.use('/api/v1/upload', uploadRoute);

// app.post('/api/v1/upload', upload.single('file'), (req, res) => {

//   res.redirect('/');

// });

app.get('/api/v1/upload', (req, res) => {

  console.log();

  gfs.files.find().toArray((err, files) => {

    if (err) {

      return res.status(500).json({ error: 'Failed to get files' });

    }

    return res.json(files);

  });

});

//=========file/

const PORT = 3000;

http.createServer({}, app).listen(PORT);

console.log(`Server running at ${PORT}`);

FIELES.JS ROUTES

const express = require('express');

const router = express.Router();

// Обработчик маршрута загрузки файла

router.post('/', (req, res) => {

  // Здесь вы можете добавить код для сохранения файла в MongoDB с использованием Mongoose GridFS

  console.log(req.file); // Информация о загруженном файле

  res.status(200).send('File uploaded successfully!!!!!!!');

});

// Обработчик маршрута получения всех файлов

module.exports = router;